

1 **In the Claims**

2 Claims 1, 3-16, 18-40 are pending and are listed below:

3
4 1. (Previously Presented) A software architecture for a distributed
5 computing system comprising:

6 an application configured to handle requests submitted by remote devices
7 over a network; and

8 an application program interface to present functions used by the
9 application to access network and computing resources of the distributed
10 computing system, the application program interface comprising various types
11 related to constructing user interfaces, wherein the various types comprise:

12 classes which represent managed heap allocated data that has
13 reference assignment semantics;

14 interfaces that define a contract that other types can implement;

15 delegates that are object oriented function pointers;

16 structures that represent static allocated data that has value
17 assignment semantics; and

18 enumerations which are value types that represent named constants.

19
20 2. (Canceled).

21
22 3. (Original) A software architecture as recited in claim 1, wherein
23 the distributed computing system comprises client devices and server devices that
24 handle requests from the client devices, the remote devices comprising at least
25 one client device.

1
2 4. (Original) A software architecture as recited in claim 1, wherein
3 the distributed computing system comprises client devices and server devices that
4 handle requests from the client devices, the remote devices comprising at least
5 one server device that is configured as a Web server.

6
7 5. (Previously Presented) An application program interface embodied
8 on one or more computer readable media, comprising: multiple types related to
9 constructing user interfaces, the types comprising classes which represent
10 managed heap allocated data that has reference assignment semantics, interfaces
11 that define a contract that other types can implement, delegates that are object
12 oriented function pointers, structures that represent static allocated data that has
13 value assignment semantics and enumerations which are value types that
14 represent named constants.

15
16 6. (Original) An application program interface as recited in claim 5,
17 wherein the classes comprise a forms class that represents a window or a dialog
18 box that makes up an application's user interface.

19
20 7. (Original) An application program interface as recited in claim 6,
21 wherein the forms class has multiple members comprising one or more of: public
22 static properties, public static methods, public instance constructors, public
23 instance methods, public instance properties, public instance events, protected
24 instance properties, and protected instance methods.

1 8. (Original) An application program interface as recited in claim 5,
2 wherein the type comprising the interfaces comprises a button control interface
3 that allows a control to act like a button on a form.
4

5 9. (Original) An application program interface as recited in claim 5,
6 wherein the type comprising the interfaces comprises a container control interface
7 that provides functionality for a control to act as a parent for other controls.
8

9 10. (Original) An application program interface as recited in claim 5,
10 wherein the type comprising the interfaces comprises an editing notification
11 interface.
12

13 11. (Original) An application program interface as recited in claim 5,
14 wherein the type comprising the interfaces comprises a data object interface that
15 provides a format independent mechanism for transferring data.
16

17 12. (Original) An application program interface as recited in claim 5,
18 wherein the type comprising the interfaces comprises a feature support interface
19 that specifies a standard interface for retrieving feature information from a current
20 system.
21

22 13. (Original) An application program interface as recited in claim 5,
23 wherein the type comprising the interfaces comprises a message filter interface.
24
25

14. (Original) An application program interface as recited in claim 5,
wherein the type comprising the interfaces comprises a handle-exposing interface
to expose handles.

15. (Original) An application program interface as recited in claim 5,
wherein the type comprising the interfaces comprises one or more of the
following interfaces:

a button control interface that allows a control to act like a button on a
form;

a container control interface that provides functionality for a control to act
as a parent for other controls;

an editing notification interface;

a data object interface that provides a format independent mechanism for
transferring data;

a feature support interface that specifies a standard interface for retrieving
feature information from a current system;

a message filter interface; and

a handle-exposing interface to expose handles.

16. (Previously Presented) A distributed computer software
architecture, comprising:

one or more applications configured to be executed on one or more
computing devices, the applications handling requests submitted from remote
computing devices;

a networking platform to support the one or more applications; and

1 an application programming interface to interface the one or more
2 applications with the networking platform, the application programming interface
3 comprising various types related to constructing user interfaces, wherein the
4 various types comprise:

5 classes which represent managed heap allocated data that has reference
6 assignment semantics;

7 interfaces that define a contract that other types can implement;

8 delegates that are object oriented function pointers;

9 structures that represent static allocated data that has value assignment
10 semantics; and

11 enumerations which are value types that represent named constants.

12
13 17. (Canceled).

14
15 18. (Previously Presented) A distributed computer software
16 architecture as recited in claim 16, wherein the classes comprises a forms class
17 that represents a window or a dialog box that makes up an application's user
18 interface.

19
20 19. (Original) A distributed computer software architecture as recited
21 in claim 18, wherein the forms class has multiple members comprising one or
22 more of: public static properties, public static methods, public instance
23 constructors, public instance methods, public instance properties, public instance
24 events, protected instance properties, and protected instance methods.

1 20. (Previously Presented) A distributed computer software
2 architecture as recited in claim 16, wherein the type comprising the interfaces
3 comprises a button control interface that allows a control to act like a button on a
4 form.

5
6 21. (Previously Presented) A distributed computer software
7 architecture as recited in claim 16, wherein the type comprising the interfaces
8 comprises a container control interface that provides functionality for a control to
9 act as a parent for other controls.

10
11 22. (Previously Presented) A distributed computer software
12 architecture as recited in claim 16, wherein the type comprising the interfaces
13 comprises an editing notification interface.

14
15 23. (Previously Presented) A distributed computer software
16 architecture as recited in claim 16, wherein the type comprising the interfaces
17 comprises a data object interface that provides a format independent mechanism
18 for transferring data.

19
20 24. (Previously Presented) A distributed computer software
21 architecture as recited in claim 16, wherein the type comprising the interfaces
22 comprises a feature support interface that specifies a standard interface for
23 retrieving feature information from a current system.

1 25. (Previously Presented) A distributed computer software
2 architecture as recited in claim 16, wherein the type comprising the interfaces
3 comprises a message filter interface.

4
5 26. (Previously Presented) A distributed computer software
6 architecture as recited in claim 16, wherein the type comprising the interfaces
7 comprises a handle-exposing interface to expose handles.

8
9 27. (Previously Presented) A distributed computer software architecture
10 as recited in claim 16, wherein the type comprising the interfaces comprises one
11 or more of the following interfaces:

12 a button control interface that allows a control to act like a button on a
13 form;

14 a container control interface that provides functionality for a control to act
15 as a parent for other controls;

16 an editing notification interface;

17 a data object interface that provides a format independent mechanism for
18 transferring data;

19 a feature support interface that specifies a standard interface for retrieving
20 feature information from a current system;

21 a message filter interface; and

22 a handle-exposing interface to expose handles.

23
24 28. (Previously Presented) A computer system including one or more
25 microprocessors and one or more software programs, the one or more software

1 programs utilizing an application program interface to request services from an
2 operating system, the application program interface including separate commands
3 to request services comprising services related to constructing user interfaces,
4 wherein the application program interface groups API functions into multiple
5 namespaces that define a collection of classes which represent managed heap
6 allocated data that has reference assignment semantics, interfaces that define a
7 contract that other types can implement, delegates that are object oriented
8 function pointers, enumerations which are value types that represent named
9 constants and structures that represent static allocated data that has value
10 assignment semantics.

11
12 29. (Previously Presented) A method, comprising:
13 managing network and computing resources for a distributed computing
14 system; and

15 exposing a set of functions that enable developers to access the network
16 and computing resources of the distributed computing system, the set of functions
17 comprising functions to facilitate construction of user interfaces, wherein the
18 functions are grouped into multiple namespaces that define a collection of classes
19 which represent managed heap allocated data that has reference assignment
20 semantics, interfaces that define a contract that other types can implement,
21 delegates that are object oriented function pointers, enumerations which are value
22 types that represent named constants and structures that represent static allocated
23 data that has value assignment semantics.

30. (Original) A method as recited in claim 29, further comprising receiving a request from a remote computing device, the request containing a call to the set of functions.

31. (Previously Presented) A method, comprising creating a namespace with functions that enable drawing and construction of user interfaces, the name space defining classes which represent managed heap allocated data that has reference assignment semantics, interfaces that define a contract that other types can implement, delegates that are object oriented function pointers, structures that represent static allocated data that has value assignment semantics, and enumerations which are value types that represent named constants.

32. (Original) A method as recited in claim 31, wherein the namespace defines a forms class that represents a window or a dialog box that makes up an application's user interface.

33. (Original) A method as recited in claim 32, wherein the forms class has multiple members comprising one or more of: public static properties, public static methods, public instance constructors, public instance methods, public instance properties, public instance events, protected instance properties, and protected instance methods.

34. (Original) A method as recited in claim 31, wherein the namespace defines an interface comprising a button control interface that allows a control to act like a button on a form.

1
2 35. (Original) A method as recited in claim 31, wherein the namespace
3 defines an interface comprising a container control interface that provides
4 functionality for a control to act as a parent for other controls.

5
6 36. (Original) A method as recited in claim 31, wherein the namespace
7 defines an interface comprising an editing notification interface.

8
9 37. (Original) A method as recited in claim 31, wherein the namespace
10 defines an interface comprising a data object interface that provides a format
11 independent mechanism for transferring data.

12
13 38. (Original) A method as recited in claim 31, wherein the namespace
14 defines an interface comprising a feature support interface that specifies a
15 standard interface for retrieving feature information from a current system.

16
17 39. (Original) A method as recited in claim 31, wherein the namespace
18 defines an interface comprising a message filter interface.

19
20 40. (Original) A method as recited in claim 31, wherein the namespace
21 defines an interface comprising a handle-exposing interface to expose handles.
22
23
24
25